AMENDMENTS TO THE CLAIMS

1.   (Currently Amended) A method of ~~testing~~ using a computer system to test a product configuration for configuration errors, wherein the product configuration is stored as electronic data in a computer system for generating product configurations, the computer system including at least one rule defining a relationship between at least two parts, the product configuration including a plurality of parts, the method comprising:

    entering a test case into the computer system to detect configuration errors in the product configuration, wherein the test case includes data to ~~selects at least one part to include in~~ change the product configuration; ~~and~~

    processing the test case with the computer system in accordance with the at least one rule to ~~determine~~ detect whether the change in the product configuration, as a result of processing the test case in accordance with the at least one rule, produced a configuration error ~~at least one part selected in the test case conflicts with the plurality of parts previously included~~ in the product configuration; and

    generating explanation data with the computer system to provide an explanation of any detected configuration error in the product configuration.

2.   (Currently amended) The method, as set forth in claim 1, wherein processing the ~~at least one rule to determine whether the at least one part selected in the test case conflicts with the plurality of parts previously included in the product configuration~~ test case, further includes:

    initializing the computer system with a part state;

    inputting ~~the~~ at least one part selection to change the product configuration; and

    listening to state change events in the system to detect when a state change event occurs that results in the computer system being in the initialized part state.

1   3.   (Currently amended) The method, as set forth in claim 2, wherein ~~processing the~~

2   ~~at least one rule to determine whether the at least one part selected in the test case conflicts with~~

3   ~~the plurality of parts previously included in the product configuration~~ <u>generating explanation</u>

4   <u>data</u>, further includes:

5   generating ~~a cause~~ <u>explanation data</u> that explains the part state in terms of the state

6   change event.

1   4.   (Currently amended) The method, as set forth in claim 3, wherein processing the

2   ~~at least one rule to determine whether the at least one part selected in the test case conflicts with~~

3   ~~the plurality of parts previously included in the product configuration~~ <u>test case</u>, further includes:

4   generating a new part state for each part associated with the ~~cause~~<u>change in the product</u>

5   <u>configuration</u>.

1   5.   (Currently amended) The method, as set forth in claim 4,~~wherein processing the~~

2   ~~at least one rule to determine whether the at least one part selected in the test case conflicts with~~

3   ~~the plurality of parts previously included in the product configuration~~, <u>test case</u> further includes:

4   determining ~~the~~causes that explain the new part states in terms of the state change event.

1   6.   (Currently amended) The method, as set forth in claim 5, <u>wherein generating</u>

2   <u>explanation data</u> further ~~comprising~~<u>comprises</u>:

3   generating a cause tree wherein the root of the cause tree is the initial part state, and

4   leaves of the tree are the user's selections of parts.

1   7.   (Currently amended) The method, as set forth in claim 6 <u>wherein generating</u>

2   <u>explanation data comprises</u>, further ~~comprising~~ <u>comprises</u>:

3   generating an explanation of the part state wherein the part selections are the root of the

4   explanation<u>data</u> and the causes follow from the part selections.

1   8.   (Currently amended) The method, as set forth in claim ~~7~~<u>1</u>, wherein the

2   explanation<u>data</u> is based on selection of a part.

1 9. (Currently amended) The method, as set forth in claim ~~7~~1, wherein the
2 explanation data is based on execution of a rule.

1 10. (Currently amended) The method, as set forth in claim ~~7~~1, wherein the
2 explanation data is based on a part being in two states at the same time.

1 11. (Currently amended) The method, as set forth in claim ~~7~~1, wherein the
2 explanation data is based on a requires choice rule that cannot be satisfied.

1 12. (Currently amended) The method, as set forth in claim ~~7~~1, wherein the
2 explanation data is based on a look ahead process.

1 13. (Original) The method, as set forth in claim 7, further comprising:
2 sorting the tree by iteration number, wherein the iteration number of a part state is
3. determined by measuring the longest distance between the part state and the cause
4 corresponding to the part state.

1 14. (Currently amended) ~~An article of manufacture~~ A computer program product
2 having code embodiment therein to cause a processor to test a product configuration for
3 configuration errors, wherein the product configuration is stored as electronic data in a computer
4 system, the computer system including at least one rule defining a relationship between at least
5 two parts, the product configuration including a plurality of parts, the code comprising:
6 ~~a computer usable medium having computer readable program code embodied therein for~~
7 ~~testing a product configuration in a system for generating product configurations,~~
8 ~~the system including at least one rule defining a relationship between at least two~~
9 ~~parts, the product configuration including a plurality of parts, the computer~~
10 ~~readable program code including:~~
11 computer readable program code configured to cause ~~a~~ the computer system to
12 allow a user to enter a test case into the computer system to detect
13 configuration errors in the product configuration, wherein the test case

14   includes data to ~~selects at least one part to include in~~ change the product

15   configuration; ~~and~~

16   computer readable program code configured to cause ~~a~~ the computer system to

17   process the test case with the computer system in accordance with the at

18   least one rule to ~~determine~~ detect whether the change in the product

19   configuration, as a result of processing the test case in accordance with the

20   at least one rule, produced a configuration error~~at least one part selected in~~

21   ~~the test case conflicts with the plurality of parts previously included in the~~

22   ~~product configuration.~~; and

23   computer readable program code configured to cause the computer system to

24   generate explanation data with the computer system to provide an

25   explanation of any detected configuration error in the product

26   configuration.

1   15.   (Currently amended) The article of manufacture, as set forth in claim 14, further

2   including:

3   computer readable program code configured to cause ~~a~~ the computer system to initialize

4   the computer system with a part state;

5   computer readable program code configured to cause ~~a~~ the computer system to input ~~the~~

6   at least one part selection to change the product configuration; and

7   computer readable program code configured to cause ~~a~~ the computer system to listen to

8   state change events in the system to detect when a state change event occurs that

9   results in the system being in the initialized part state.

1   16.   (Currently amended) The article of manufacture, as set forth in claim 15, further

2   including:

3   computer readable program code configured to cause ~~a computer~~ the computer system to

4   generate ~~a cause~~ explanation data that explains the part state in terms of the state

5   change event.

1   17.   (Currently amended) The article of manufacture, as set forth in claim 16, further
2   including:
3   computer readable program code configured to cause ~~a computer~~ the computer system to
4   generate a new part state for each part associated with the ~~cause~~ change in the
5   product configuration.

1   18.   (Currently amended) The article of manufacture, as set forth in claim 17, further
2   including:
3   computer readable program code configured to cause ~~a computer~~ the computer system to
4   determine ~~the~~ causes that explain the new part states in terms of the state change
5   event.

1   19.   (Currently amended) The article of manufacture, as set forth in claim 18, further
2   comprising:
3   computer readable program code configured to cause ~~a computer~~ the computer system to
4   generate a cause tree wherein the root of the cause tree is the initial part state, and
5   leaves of the tree are the user's selections of parts.

1   20.   (Currently amended) The article of manufacture, as set forth in claim 19, further
2   comprising:
3   computer readable program code configured to cause ~~a computer~~ the computer system to
4   generate an explanation of the part state wherein the part selections are the root of
5   the explanation and the causes follow from the part selections.

1   21.   (Currently amended) The article of manufacture, as set forth in claim ~~20~~14,
2   wherein the explanation data is based on selection of a part.

1   22.   (Currently amended) The article of manufacture, as set forth in claim ~~20~~14,
2   wherein the explanation data is based on execution of a rule.

1   23.   (Currently amended) The article of manufacture, as set forth in claim ~~20~~14,

2 wherein the explanation data is based on a part being in two states at the same time.

1   24.   (Currently amended) The article of manufacture, as set forth in claim ~~20~~14,

2 wherein the explanation data is based on a requires a choice rule that cannot be satisfied.

1   25.   (Currently amended) The article of manufacture, as set forth in claim ~~20~~14,

2 wherein the explanation data is based on a look ahead process.

1   26.   (Currently amended) The article of manufacture, as set forth in claim 20, further

2 comprising:

3   computer readable program code configured to cause ~~a computer~~ the computer system to

4   sort the tree by iteration number, wherein the iteration number of a part state is

5   determined by measuring the longest distance between the part state and the cause

6   corresponding to the part state.

1   27.   (Currently amended) An apparatus for testing a product configuration for

2 configuration errors generated by a product configuration system, comprising:

3   a memory having stored therein at least one rule defining a relationship between at least

4   two parts in the product configuration;

5   a test case to detect configuration errors in the product configuration, wherein the test

6   case includes data to change the product configuration ~~pertaining to at least one~~

7   ~~part to include in the product configuration~~; and

8   a processor coupled to the memory to (a) process ~~receive~~ the at least one rule and the test

9   case, ~~wherein the processor is operable to determine~~ (b) detect whether the change

10   in the product configuration, as a result of processing the test case in accordance

11   with the at least one rule, produced a configuration error ~~at least one part in the~~

12   ~~test case conflicts with the plurality of parts previously included in the product~~

13   ~~configuration according to the at least one rule~~ and (c) generate explanation data

14   to provide an explanation of any detected configuration error in the product

15   configuration.

1   28.   (Currently amended) The apparatus, as set forth in claim 27, wherein the

2   processor is further operable to:

3   initialize the configuration system with a part state;

4   to input the at least one part selection to change the product configuration;

5   to listen to state change events in the system; and

6   to detect when a state change event occurs that results in the configuration system being

7   in the initialized part state.

1   29.   (Currently amended) The apparatus, as set forth in claim 28, wherein the

2   processor is further operable to:

3   generate a cause explanation data that explains the part state in terms of the state change

4   event.

1   30.   (Currently amended) The apparatus, as set forth in claim 29, wherein the

2   processor is further operable to:

3   generate a new part state for each part associated with the cause change in the product

4   configuration.

1   31.   (Original) The apparatus, as set forth in claim 30, wherein the processor is further

2   operable to:

3   generate a cause tree wherein the root of the cause tree is the initial part state, and leaves

4   of the tree are the user's selections of parts.

1   32.   (Original) The apparatus, as set forth in claim 30, wherein the processor is further

2   operable to:

3   generate an explanation of the part state wherein the part selections are the root of the

4   explanation and the causes follow from the part selections.

1   33.   (Currently amended) The apparatus, as set forth in claim 3227, wherein the

2   explanation data is based on execution of a rule.

1   34.   (Currently amended) The apparatus, as set forth in claim ~~32~~ 27, wherein the

2 explanation data is based on a part being in two states at the same time.

1   35.   (Currently amended) The apparatus, as set forth in claim ~~32~~ 27, wherein the

2 explanation data is based on a requires a choice rule that cannot be satisfied.

1   36.   (Currently amended) The apparatus, as set forth in claim ~~32~~ 27, wherein the

2 explanation data is based on a look ahead process.

1   37.   (Original) The apparatus, as set forth in claim 30, wherein the processor is further

2 operable to:

3   sort the tree by iteration number, wherein the iteration number of a part state is

4   determined by measuring the longest distance between the part state and the cause

5   corresponding to the part state.

1   38.   (Canceled).

1   39.   (Currently amended) The ~~configuration system of claim 38~~ apparatus as set forth

2 in claim 27, wherein the test case further includes ~~the~~ a product selection.

1   40.   (Currently amended) The ~~configuration system of claim 38~~ apparatus as set forth

2 in claim 27 wherein the product configuration comprises~~, further comprising:~~ at least one vector,

3 wherein said vector comprises a bit field, further wherein the bit field comprises bits that

4 represent elements in a configuration.

1   41.   (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in

2 claim 40, wherein the number of bits in the bit field is equal to the total number of elements and

3 an element's bit can be set or reset to specify that state of the element in the configuration.

1   42.   (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in

2   claim 40, wherein the vector specifies whether an element has been selected by the user during

3   the configuration.

1   43.   (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in

2   claim 40, wherein excluded vectors identify whether an element is excluded from a

3   configuration.

1   44.   (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in

2   claim 40, wherein removed vectors identify whether an element is removed from a configuration.

1   45.   (Currently amended) The ~~configuration system of claim~~ apparatus as set forth in

2   claim 40, wherein the vector identifies whether an element is selectable.

1.   46.   (Currently Amended) A ~~database~~ The apparatus as set forth in claim 40 further

2   comprising:

3   a database having at least one table, wherein said table represents relationships between

4   elements in a configuration~~;~~ and having at ~~lease~~ least one modified rule, wherein

5   the rule is modified based on ~~the~~ results of testing a product selection.

1   47.   (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46,

2   wherein said table represents "includes" relationships between elements in a configuration.

1   48.   (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46,

2   wherein said table represents "excludes" relationships between elements in a configuration.

1   49.   (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46,

2   wherein said table represents "removes" relationships between elements in a configuration.

1   50.   (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46,

2 wherein said table represents "requires choice" relationships between elements in a

3 configuration.

1   51.   (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 50,

2 wherein the representation of "requires choice" relationships includes a pointer to a group table

3 that includes a bit vector that identifies the elements that are contained in the group from which a

4 choice is to be made.

1   52.   (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 50,

2 wherein the representation of "requires choice" relationships includes minimum and maximum

3 designations to identify the minimum and maximum number of group members that are to be

4 selected to satisfy the "requires choice" relationship.

1   53.   (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 46,

2 wherein said table includes a left-hand side and a right-hand side.

1   54.   (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 53,

2 wherein the left-hand side includes a bit vector that contains bits corresponding to elements.

1   55.   (Currently amended) The ~~database of claim~~ apparatus as set forth in claim 53,

2 wherein the right-hand side includes one or more bit vectors that represent configuration

3 elements.

1   56.   (Currently amended) <u>The apparatus as set forth in claim 27 wherein the test case</u>

2 <u>further comprises data representing</u> ~~A test case for testing a product configuration generated by a~~

3 ~~product configuration system, comprising~~:

4   a product selection;

5   at least one part selection; and

6   an expected state of the selected part based on one or more rules.

Serial No. 09/773,101

1   57.   (Currently amended) ~~A method for identifying an invalid configuration generated~~
2   ~~by a product configuration system, comprising:~~ The method as set forth in claim 1 wherein the
3   test case further comprises data to:
4   ~~selecting~~ select a product;
5   ~~selecting~~ select at least one part; and
6   ~~generating~~ generate a part state of the selected part based on one or more rules.

1   58.   (Canceled).

1   59.   (Canceled).

1   60.   (Canceled).

1   61.   (Original) The method as set forth in claim 57, further comprising:
2.   determining whether the product is selectable.

1   62.   (Canceled).

1   63.   (Original) The method as set forth in claim 57, further comprising:
2   reporting the state of the product as not selectable when selection of the product would
3   conflict with the rule.

1   64.   (Original) The method as set forth in claim 57, further comprising:
2   determining sets of parts that are excluded or deleted based on the product.

1   65.   (Currently amended) The method as set forth in claim 57, further comprising:
2   detecting when a state change event occurs that results in the computer system being in
3   the initialized part state.

1   66.   (Canceled).

1   67.   (Canceled).

1   68.   (Canceled).

1   69.   (Canceled).

1   70.   (Currently amended) An apparatus for testing a product configuration for
2   configuration errors generated by a computer implemented product configuration system,
3   comprising:
4   means for defining a relationship between at least two parts in the product configuration;
5   means for defining a test case for at least one part to include in the product configuration
6   to detect configuration errors in the product configuration, wherein the test case
7   includes data to change the product configuration; and
8   means for determining whether the at least one part in the test case conflicts with the
9   plurality of parts previously included in the product configuration according to at
10   least one rule
11·   means for processing the test case with the product configuration system in accordance
12   with the at least one rule to detect whether the change in the product
13   configuration, as a result of processing the test case in accordance with the
14   relationship between at least two parts in the product configuration, produced a
15   configuration error in the product configuration; and
16   means for generating explanation data with the product configuration system to provide
17   an explanation of any detected configuration error in the product configuration.

1   71.   (Currently amended) The apparatus, as set forth in claim 70, further comprising:
2   means for initializing the configuration system with a part state;
3   means for detecting a state change event in the configuration system; and
4   means for detecting when a state change event occurs that results in the configuration
5   system being in the initialized part state.

1   72.   (Original) The apparatus, as set forth in claim 71, further comprising:
2   means for generating a cause that explains the part state in terms of the state change
3   event.

1   73.   (Original) The apparatus, as set forth in claim 72, further comprising:

2   means for generating a new part state for each part associated with the cause.

1   74.   (Original) The apparatus, as set forth in claim 73, further comprising:

2   means for generating a cause tree, wherein the root of the cause tree is the initial part

3   state, and leaves of the tree are the user's selections of parts.

1   75.   (Original) The apparatus, as set forth in claim 73, further comprising:

2   means for generating an explanation of the part state, wherein the part selections are the

3 root of the explanation and the causes follow from the part selections.

1   76.   (Original) The apparatus, as set forth in claim 70, further comprising:

2   means for modifying the at least one rule when the test case conflicts with the plurality of

3 parts previously included in the product configuration.

1   77.   (New) The method, as set forth in claim 1, wherein the test case further includes

2 data to select at least one part to include in the product configuration and processing test case

3 further comprises:

4   processing the at least one rule to determine whether the at least one part selected in the

5   test case conflicts with the plurality of parts previously included in the product

6   configuration.

1   78.   (New) The computer program product, as set forth in claim 14, wherein the test

2 case further includes data to select at least one part to include in the product configuration and

3 the computer readable program code configured to cause the computer system to process the test

4 case further comprises:

5   computer readable code to process the at least one rule to determine whether the at least

6   one part selected in the test case conflicts with the plurality of parts previously

7   . included in the product configuration.

1   79.   (New) The apparatus, as set forth in claim 27, wherein the test case further

2   pertains to including at least one part in the product configuration and the processor is further

3   operable to:

4   determine whether the at least one part in the test case conflicts with the plurality of parts

5   previously included in the product configuration according to the at least one rule.

1   80.   (New) The apparatus, as set forth in claim 70, wherein the test case is further

2   defined to include at least one part in the product configuration and the means for processing the

3   test case includes:

4   means for determining whether the at least one part in the test case conflicts with the

5   plurality of parts previously included in the product configuration according to the at least one

6   rule.